1 Objectives

The topic for this lecture is a kind of grammar that works well with recursive-descent parsing.

- Know how to tell if a grammar is LL.
- Know what parsing technique will work with an LL grammar.
- Know how to detect and eliminate left recursion.
- Know how to detect and eliminate common prefixes.

Further reading: See Dragon Book §4.x

2 Problems

Here are some problems that will help you study for the exam. Some of these may be done in class.

1. One of these is LL, the other two are not. Fix the ones that are not.

Grammar 1	**Grammar 2**	**Grammar 3**
$E \rightarrow E\ x\ y$	$S \rightarrow A\ x$	$S \rightarrow A\ x$
$E \rightarrow E\ x\ B$	$S \rightarrow B\ y$	$S \rightarrow B\ y$
$E \rightarrow q$	$A \rightarrow z\ B$	$A \rightarrow z\ B$
$B \rightarrow E\ z$	$B \rightarrow w\ A$	$B \rightarrow z$

2. The following grammar is LL. Write some OCaml code to implement a parser for it.

$$
\begin{aligned}
S &\rightarrow v\ e\ E \\
E &\rightarrow +\ E\ E \\
E &\rightarrow *\ E\ E \\
E &\rightarrow i
\end{aligned}
$$

3. What is the definition of an LL grammar?
4. Is every LL grammar context-free? Is every context-free grammar LL?
5. What is the competitive advantage of an LL grammar?